Akraino KNI blueprints

https://wiki.akraino.org/display/AK/Kubernetes-Native+Infrastructure+for+Edge+%28KNI-Edge%29+Family

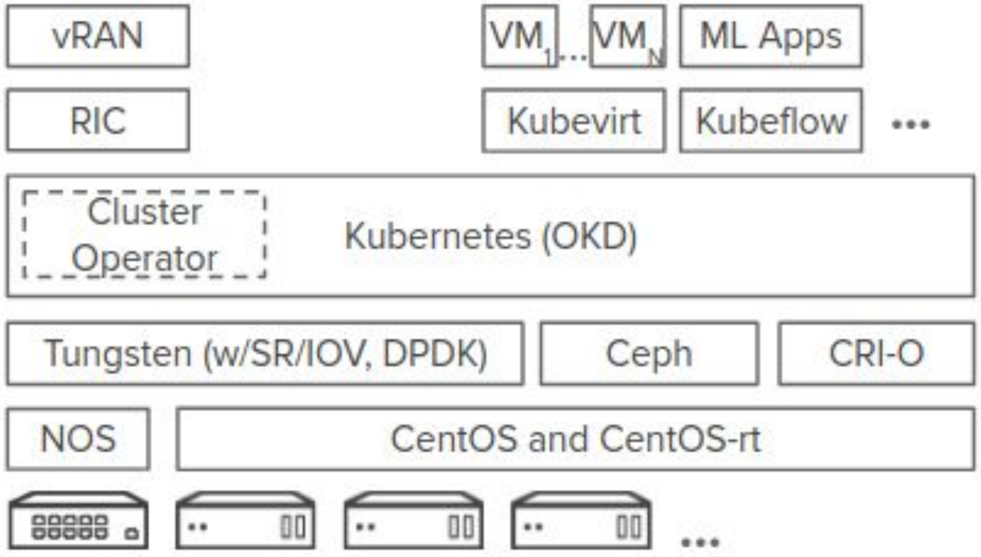
- Deploy the whole Edge Stack based on Kubernetes
- Allow users to declaratively configure, deploy and manage lifecycle of Kubernetes clusters
  - Based on Kubernetes Cluster API
  - Independent of provider: on-prem or public cloud, VMs or baremetal
- Use Operator Framework for application deployment, lifecycle management, scale, RBAC control
- Optimized for workloads on Kubernetes containers, but will allow VMs with KubeVirt as needed

AKRAINO
EDGE STACK

KNI provider access edge blueprint

https://wiki.akraino.org/display/AK/Provider+Access+Edge+%28PAE%29+Blueprint

- Industry sector: telco and carrier networks
- Example use case: mobile gaming
- Footprint: minimum 3 servers, up to 1 rack
- Workloads:
    - vRAN (OpenAirInterface)
    - MEC platform
    - AI/ML platform (KubeFlow)
- Settings:
    - App lifecycle management: Kubernetes operators
    - Cluster lifecycle management: Cluster API / controllers
    - Monitoring: Prometheus
    - Container platform: Kubernetes (OKD 4.0)
    - Container runtime: CRI-O
    - VM runtime: Kubevirt
    - OS: RHEL CoreOS, CentOS-RT
    - SDN: Tungsten Fabric (w/ SR-IOV, DPDK, and multi-network via Multus); leaf-and-spine fabric mgmt.
    - SDS: Ceph

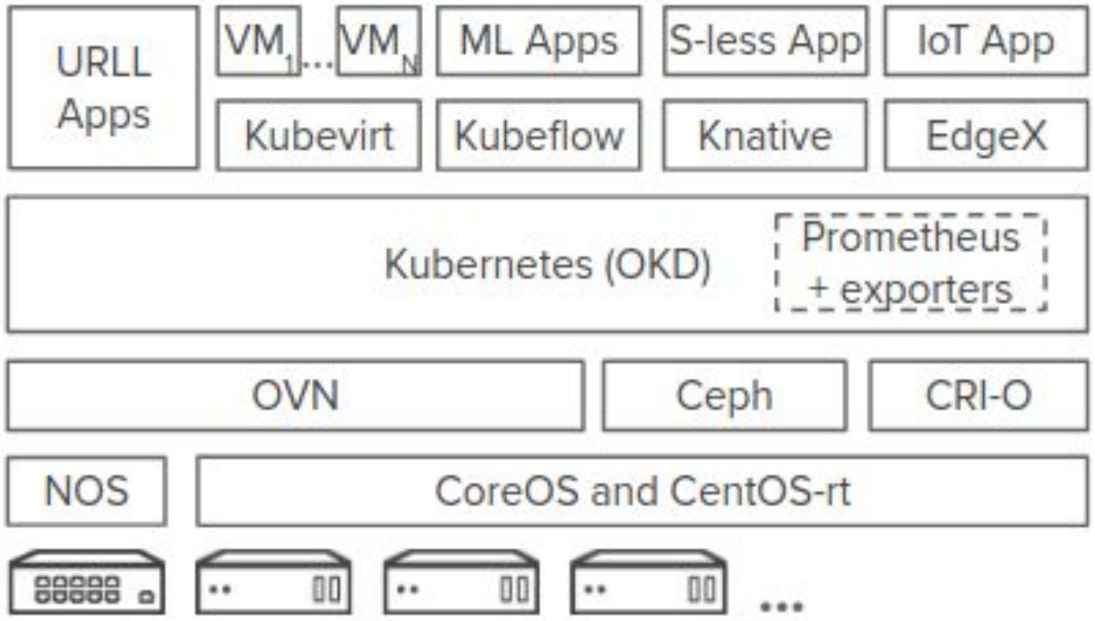


AKRAINO EDGE STACK

KNI industrial edge blueprint

https://wiki.akraino.org/display/AK/Industrial+Edge+%28IE%29+Blueprint

- Industry sector: manufacturing, energy
- Example use case: manufacturing process quality control
- Footprint: minimum 3 servers, up to 1 rack
- Workloads:
  - AI/ML (KubeFlow)
  - Serverless (Knative)
  - IoT (EdgeX?)
- Settings:
  - App lifecycle management: Kubernetes operators
  - Cluster lifecycle management: Cluster API / controllers
  - Monitoring: Prometheus
  - Container platform: Kubernetes (OKD 4.0)
  - Container runtime: CRI-O
  - VM runtime: Kubevirt
  - OS: RHEL CoreOS
  - SDN: OVN
  - SDS: Ceph



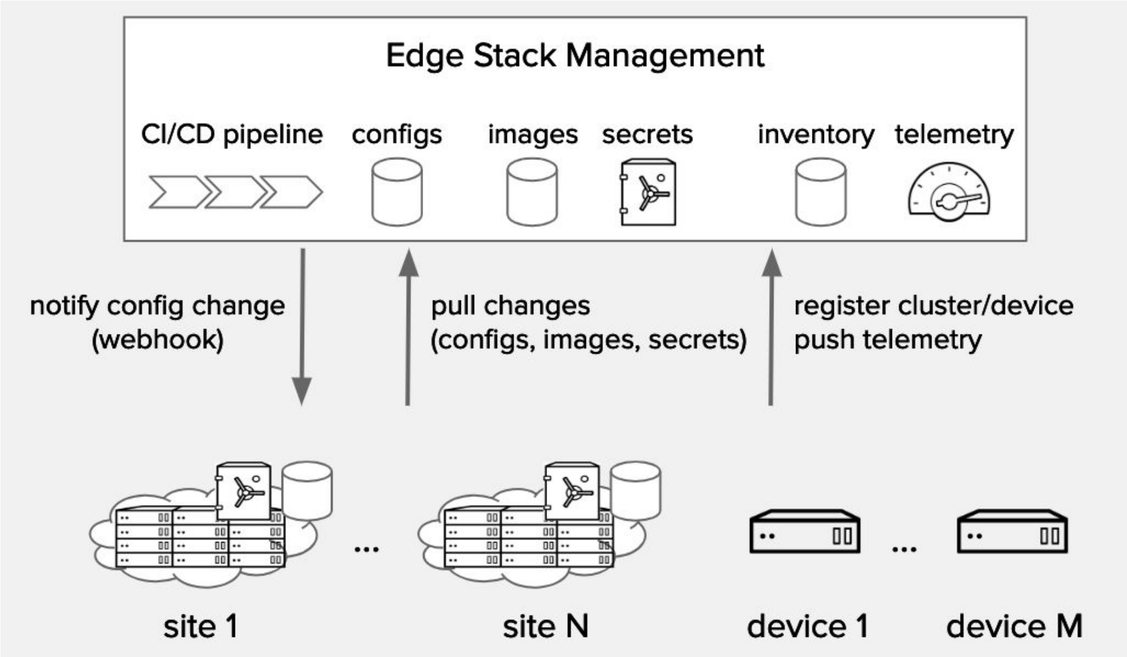
Container platform: Kubernetes (OKD 4.0)

https://github.com/openshift/installer

- Previously Ansible-based installer, now fully Operator-/Controller-based
- Easy installer: download a binary, complete wizard questions, deploy cluster
- Supported platforms: AWS, libvirt (development), baremetal (WIP)
- Based on Cluster API, with a Provider for each platform
- Introduces the concept of MachineSet and Machine object definition
- No need to provision your own hosts for Kubernetes cluster, installer will do it for you
- Workflow:
  - Ignition files for bootstrap and master/workers are generated, as well as Kubernetes manifests
  - Installer provisions the initial infrastructure: bootstrap VM, nodes for master/workers, network… using Terraform
  - Bootstrap vm starts along with his ignition file, creates a temporary cluster with MachineConfigServer endpoint
  - Master nodes are booted, communicate with MachineConfigServer and retrieve their ignition config
  - When all masters have joined the temporary cluster, they form a final etcd cluster, then they form a final production control plane
  - Kubernetes specific components are injected on production control plane: Machine API Operator
  - Workers are added to production control plane using this Machine API operator
  - Bootstrap node is removed, extra applications are deployed through operators
  - Success is reported, kubeconfig file is populated as well as console credentials

KNI blueprints architecture

- Primary need: address scalability, resiliency, connectivity
- All edge cluster/devices will be fully and self-provisioning deployments
- Each cluster is defined in a declarative way and stored on a git repo
- On provisioning, clusters/devices pull artifacts from HTTP endpoints
- Caching can be enabled to reduce load and prevent connectivity issues
- Changes on git repo will be notified with hooks, clusters will poll regularly for changes in artifacts
- On changes detected, modifications are applied on cluster


Edge Stack Management
CI/CD pipeline
configs
images
secrets
inventory
telemetry
notify config change (webhook)
pull changes (configs, images, secrets)
register cluster/device push telemetry
site 1
...
site N
device 1
...
device M

How to start working (WIP)

- Current project:
  - https://github.com/redhat-nfvpe/kni-edge-base (common docs/tools; to be moved to Akraino git)
  - https://gerrit.akraino.org/r/gitweb?p=kni/blueprint-pae.git;a=tree, https://gerrit.akraino.org/r/gitweb?p=kni/blueprint-ie.git;a=tree (blueprints)
- Two providers: libvirt, aws. Sample deployments for 1 master/worker, 3 master/workers
- Baremetal still WIP, just documented manually
- Pre-requisites:
  - Install the dependencies for libvirt / AWS as documented
  - Create an account in cloud.openshift.com and download pull secret https://cloud.openshift.com/clusters/install . You can get oc client as well for later
  - Create a private repo in github with those files: coreos-pull-secret, ssh-pub-key, [aws-access-key-id, aws-secret-access-key]
  - Create a settings.yaml with your cluster definition. Samples are provided for libvirt and AWS
  - For libvirt, compile your openshift-install binary following provided instructions
- Deploy cluster using targets in Makefile for the chose option:

```
make [aws-1-node | aws-3-node | libvirt-1-node | libvirt-3-node] CREDENTIALS=<path_to_private_repo> SETTINGS=file:///<path_to_sample_settings>
[INSTALLER_PATH=file:///<path_to_installer_binary>]
```

- It will generate a build directory, install artifacts and start the cluster build
- Start managing it with kubeconfig or console.

AKRAINO EDGE STACK

How to customize the cluster (WIP)

- On the build directory, execute: openshift-install create manifests command
- Two folders will be created:
  - manifests: used for first-time deploy
  - openshift: specific manifests applied after production control plane is up
- Start modification and testing cycle:
  - Remove previous cluster: make clean command
  - Generate new manifests: openshift-install create manifests
  - Modify the values you are interested on / add new manifests
  - Deploy new cluster: openshift-install create cluster
  - Wait for cluster to be up, retry!

Questions, next meetings

- ONS Summit Demo recording?
- Next meetings to be scheduled on zoom (pending)
  - best time slot?
- Chat channel (slack, IRC) ?
- Mailing list: blueprints@lists.akraino.org (add hashtag #kni at end of subject)

Some Useful References

- OKD/OpenShift:
  - https://github.com/openshift/installer/
- KubeVirt:
  - http://kubevirt.io/get_kubevirt/ (quickstart)
  - https://kubevirt.io/user-guide/docs/latest/welcome/index.html (documentation)
- Operators:
  - https://operators.gitbook.io/operatorhub/ (introduction)
  - https://github.com/operator-framework/operator-sdk (SDK for creating new Operators)
  -